Issue Date: FoxTalk August 2000

Demystifying Visual FoxPro Report Files

John S. Koziol, IV

Visual FoxPro reports are stored in tables with an extension of FRX. John Koziol recently dug deep into the FRX to develop a report interpreter that could export a report to Microsoft Word 2000. Here are the results of his investigation.

Visual FoxPro stores reports in two files, one with an FRX extension and one with an FRT extension. These files are sometimes called metadata—information stored in tables that's interpreted internally by Visual FoxPro. The FRX file is the table component and can be opened like any other table; the FRT file is the equivalent of a table's FPT file: Data within memo fields in the FRX is stored in the FRT file, and the file shouldn't be opened by itself.

In order to work with reports beyond the capabilities of the native Visual FoxPro report engine, it's necessary to understand what's stored in the FRX file so that it can be interpreted. Microsoft provides reports that detail the FRX structure in a subfolder of the Visual FoxPro 6.0 home folder, called FILESPEC. These reports are named 60FRX1 and 60FRX2. 60FRX1 shows the physical structure of the FRX, a description of values per field, and the report objects that a field might apply to. 60FRX2 extends the report to show more "applies to" report objects.

"So," you're probably thinking at this point, "report definitions are well-documented, should I feel compelled to mess around with them."

Yes and no. What I've discovered is that the FRX stores report definition data in a way that requires very careful analysis and processing. Not all is as it seems within the FRX, and some field values, critical to the report even, are just plain undocumented. Others make no sense until they're placed in context with other properties. Worse yet is that it appears that the reports in the FILESPEC directory can be misleading or just plain wrong.

Note: Report file information in this article applies to Visual FoxPro versions 5.0 through 6.0 Service Pack 3. Some overview information and standards are important to gain a complete understanding of the FRX. This, mainly, is the information missing from the reports in the FILESPEC directory. Without understanding these standards, the FRX can't be interpreted.

A disclaimer should be noted: I don't have all of the answers about FRX structure. Some values in some fields remain a mystery to me.

Basic structure

Each FRX file has a single report definition record. It contains overall information about the report, as well as printer driver information for the printer in use when the report was created.

Following the report definition record are report band records. One record per band (title, page header, group header, and so forth) is in the file. Then there are report object records. Each record contains information about a single report object such as a label, a field, a shape, and so on.

Next, if used in the report, are report variable records—one per variable defined in the report. After the variable records are the font object records. It appears to contain one record per font, font size, and font style.

Following the font records is a report data environment record. This record exists whether or not the report actually uses the data environment.

Last but not least are data object records. These records contain information about the cursors and relationships defined in the data environment of the report.

FRX record order

Record order is important in the FRX file for some objects, but not for others. For example, the first record will always be the report definition record. After that, records for the report bands are stored in order of band type.

Footer bands are detailed in the reverse order of the matching header. So band records will be in the order GroupHeader1, GroupHeader2, Detail, GroupFooter2, GroupFooter1.

Records for other report objects appear to be stored in creation order after the report definition record and any band records.

General units of measure and placement

Understanding the unit of measure employed in a report file is key to understanding how Visual FoxPro interprets the file at runtime. Units of measure are used in the fields VPOS, HPOS, HEIGHT, and WIDTH (exceptions are noted in the table which follows). I like to call these units FoxPro Report Units (FRU). There are no fields in the FRX assigning objects to bands; you can only calculate item placement as detailed in the following paragraphs. In other words, you put an object somewhere on a report. Then, when you define a band's position, any objects within that band's position are "scooped up" in the band.

What's an FRU? It depends on your ruler. If your report was designed using inches as the dimension, an FRU is 1/10,000 of an inch. If you're using centimeters, an FRU is 1/1,000 of a centimeter. The default for the English language version of VFP is inches; foreign versions default to centimeters. If you go to Page Setup in the Report Designer, on the form there will be a label captioned "Dimensions:" and to it's right will be the current value.

How are these FRU used? Here's where we get to the tricky part.

In the first record—the report definition record—the only place where FRU are used is in the WIDTH field. If the report is a multiple-column report, WIDTH will contain the width of each column in FRU. This corresponds to the width settings in Page Setup. In all cases, VPOS will contain the column count.

In the report bands records, HEIGHT is the only value stored in FRU. There's a non-zero value stored in the WIDTH field in report summary band records; I've yet to determine what that value is and whether it can be ignored for measuring purposes. As to HEIGHT, what's stored there is the absolute height of the band in FRU.

Report objects—that is, anything that's not data environment information and not the report definition or report bands—use all four fields. The VPOS value is an absolute measurement, in FRU, of the first instance of the report object with the value, again in FRU, of the height of any graphical separator bands in the Report Designer added in. Huh? More on this soon.

The HPOS value is in FRU and is the leftmost position of the object relative to the left margin of the column. HEIGHT and WIDTH are pretty straightforward and measure that size of the object in FRU.

Here's where it becomes necessary to become familiar with a "magic number": the graphical band height. This "magic number" is 1979.16666666, when inches are used. When centimeters are used, the number is 502.7083333.

To better understand this, look at **Figure 1**. This is a screen shot of the Report Designer with a very simple report, called DEMO.FRX. This report was created using inches as the scale.

***Figure 1:* DEMO.FRX.**

We have three bands: a page header, a detail band, and a page footer. These bands each contain a label. **Table 1** illustrates the placement values of these six records.

***Table 1*. Placement values for bands and objects.**

Record	VPOS	HPOS	HEIGHT	WIDTH
Page header band			3646.000	
Detail band			5000.000	
Page footer band			5000.000	
Page header label	1250.000	19791.667	1666.667	35000.000

Detail label	6979.167	23750.000	1666.667	28333.333
Page footer label	15312.500	20000.000	1666.667	35000.000

Note that the report objects are listed in top-to-bottom order, but would actually be stored in the FRX in creation order.

Okay, now what? Well, VPOS and HPOS aren't used for bands, and WIDTH is determined from the report definition record and is always zero for bands. HEIGHT is the absolute height, in FRU, of the band. But simply adding together the band heights doesn't give you everything needed to determine where the report objects are placed: The graphical band height has to be added to the bottom of each band to determine a placement range for the labels. **Table 2** shows how to do this. Please note that all measurements are in FRU and that some rounding was employed.

Table 2. Determining band ranges.

Band	HEIGHT	Starts at	Start+HEIGHT	Band height	Actual end
Page header	3646.000	0.000	3646.000	1979.16666666	5625.167
Detail	5000.000	5625.167	10625.167	1979.16666666	12604.333
Page footer	5000.000	12604.333	17604.333	1979.16666666	19583.500

See? Now I've determined the actual range of each band and can determine what report objects belong to what bands, and the object's relative position in the band. Relative position is calculated as object VPOS—Band "Starts At" for vertical position. **Table 3** gives an example where HPOS is absolute to the left margin of this report as we only have a single column and is ignored.

Table 3. Calculating band and relative vertical position.

Object	VPOS	Band as calculated?	Relative VPOS
Page header label	1250.000	Header	1250.000
Detail label	6979.167	Detail	1354.000
Page footer label	15312.500	Footer	2708.167

And that's all there is to calculating what object goes into what band and where. I wish that Microsoft would have documented these calculations and the graphical band height conundrum, as this always seems to be a major stumbling block for folks trying to work directly with report metadata files.

FRX structure and value ranges

As I mentioned before, this article doesn't detail every single possible value in every single field. Frankly, I don't know them all, and I'm sure that there are some values that shall remain known only to the friendly folks on the Fox Team at Microsoft.

Table 4 is an attempt to show all of the fields in the FRX and how each type of FRX record uses those fields. Please feel free to e-mail me at i486dx@prodigy.net if you discover any omissions in this document.

Table 4. The FRX in all its glory.

Field	**Structure**	**Values**
PLATFORM	C(8)	Always "WINDOWS" in all records.
UNIQUEID	C(10)	SYS(2015)-like value. When used, each record has a unique value. Not used for font object, data environment, or data objects.
TIMESTAMP	N(10,0)	A timestamp for the last time a record was changed.
OBJTYPE	N(2,0)	The type of object that a record is describing. 1 = Report definition record 5 = Label 6 = Line 7 = Shape. Will also show OBJCODE of 4 when used for unknown reasons. 8 = Field 9 = Band record 17 = Bitmap, either standalone or General field 18 = Variable 23 = Font object 25 = Data environment record 26 = Data object, either a Cursor or Relation (NAME field contains type)
OBJCODE	N(3,0)	Used when a record with a certain OBJTYPE needs more definition. When OBJTYPE = 9 (a band), the following values are used: 0 = Title 1 = Page header 2 = Column header 3 = Group header 4 = Detail 5 = Group footer 6 = Column footer 7 = Page footer 8 = Summary In the report definition record, OBJCODE = 53. With shapes, OBJCODE = 4. For report

		objects, the value is zero. With variables, the data environment record, and data objects, this value is blank.
NAME	Memo	For report variables (OBJTYPE = 18), contains the variable name. For the data environment record, contains data "environment." For cursors and relations, contains "cursor" or "relation," respectively. The Microsoft documentation says that filename is stored here for bitmaps. Wrong.
EXPR	Memo	For the report definition record, contains printer driver and print setup information. For group headers, contains the group expression. Important note: contains nothing for group footers. If you need the group expression for a group footer, you must find the group header. For labels, contains the label caption. For fields, contains the field expression. For report variables, contains the value to store. For data environment and data object records, contains the properties for those objects; for instance, Alias = Cursor1.
VPOS	N(9,0)	In the report definition record, contains the number of columns in the report. It's left blank in band, variable, data environment, and data object records. In font object records, it appears to show the character height in pixels, as in the first FONTMETRIC () attribute. For report objects, contains position information as detailed in the article.
HPOS	N(9,0)	In the report definition record, contains the left margin in FRU. If the report has been defined for the printable page (Page Setup), this defaults to 0.000, and adjustments to this margin appear to be relative to the printable page boundaries. If the report is defined as using the whole page, this defaults to 2500.000 (a quarter inch), and the measurement is an absolute position in FRU from the physical left margin. In band, variable, data environment, and data object records, it's blank. For font object records, appears to contain the average character

		width in pixels, as in the sixth FONTMETRIC () attribute. For report object records, contains relative placement information as detailed previously.
HEIGHT	N(9,0)	Not used in the report definition for reports, but I suspect that it's used there for labels. For bands, the value is the absolute band height in FRU. For report objects, it's the absolute height of the object. For variables, the data environment, and data objects, it's left blank. For font object records, it seems to correspond to FONTMETRIC(2), character ascent in pixels.
WIDTH	N(9,0)	In the report definition record, it's the width of each column of the report. If there's only one column, it's the entire printable width of the report. For most bands, it's zero. Curiously, WIDTH contains a value for Summary bands, but I've been unable to determine the function of that value. For report objects, the value is the absolute width of the object. For variables, the data environment, and data objects, the value is blank. In font object records, it corresponds with FONTMETRIC(7), maximum character width in pixels.
STYLE	Memo	Microsoft's 60FRX1 report claims that this field contains font style and alignment information for fields and labels. This isn't the case; style information is stored in the FONTSTYLE fields and alignment is stored as a standard FoxPro picture expression in the PICTURE field. This field isn't used by any other records.
PICTURE	Memo	With report fields (OBJTYPE=8), contains a standard FoxPro picture expression. Sometimes, VFP adds a picture clause automatically when text alignment or some other settings aren't default. For bitmaps (OBJTYPE=17), contains either the General field name containing the image or the relative pathname for the picture file; say, "..\graphics\mybitmap.bmp."

ORDER	Memo	Apparently not used.
UNIQUE	Logical	For report variables, indicates whether the variable is to be released at the end of the report. According to the Microsoft 60FRX1 report, it's also set to .T. in the report definition record when a temporary report is created by a wizard. I haven't tested that. Also, according to the Microsoft report, it's used with relation records. I haven't seen a non-blank value set by any relation records, so I can't say whether this is true and what the use would be.
COMMENT	Memo	Can store developer's comments. Interestingly, a single space (ASCII code 32) is stored there by VFP for report field records.
ENVIRON	Logical	Apparently not used.
BOXCHAR	C(1)	Also not used.
FILLCHAR	C(1)	The Microsoft report claims this isn't used. Wrong. A "C" or "N" for, presumably, character or numeric information is stored for all report field records. There appears to be no use of "D," "Y," or other data types in this field, so, for example, DATE() has a FILLCHAR value of "C."
TAG	Memo	In the report definition record, contains compiled or binary printer driver information. For report variable records, contains the starting value for a variable, that is, the value a variable is reset to. The Microsoft documentation shows a use for this field with indexes and relations, but I haven't seen that.
TAG2	Memo	Contains more binary information about the printer driver in the report definition record. I haven't seen any other use for the field, contrary to Microsoft documentation.
PENRED	N(5,0)	Foreground red color value for report objects, as used in the RGB() function. You have to be careful with this, though. If no color is specified for an object, the default value is –1, which the RGB() function doesn't appreciate. For font objects, it

		appears to be the same value as FONTMETRIC(3), character descent in pixels.
PENGREEN	N(5,0)	Foreground green color value. See PENRED. For font objects, appears to be the same value as FONTMETRIC(5), extra leading in pixels.
PENBLUE	N(5,0)	Foreground blue color value. See PENRED. Not used otherwise.
FILLRED	N(5,0)	Background red color value. See PENRED.
FILLGREEN	N(5,0)	Background green color value. See PENRED.
FILLBLUE	N(5,0)	Background blue color value. See PENRED.
PENSIZE	N(5,0)	Pen size value for shapes and lines.
PENPAT	N(5,0)	Pen pattern type for shapes and lines.
FILLPAT	N(5,0)	Fill pattern value for shapes.
FONTFACE	Memo	The name of the font for report fields, labels, and font objects. The default font name for the report in the report definition record.
FONTSTYLE	N(3,0)	Where there's a FONTFACE value, there's a FONTSTYLE value. The basic FONTSTYLE values are: 0 = Normal 1 = Bold 2 = Italic 4 = Underlined 128 = Strikethrough These values may be added to one another for combined effects. For example, a value of 5 = Bold and Underlined.
FONTSIZE	N(3,0)	The size of the font for report fields, labels, and font objects. The default font size for the report definition record.
MODE	N(3,0)	Used for all report objects. A value of 1 indicates transparent, 0 is opaque.
RULER	N(1,0)	Only used in the report definition record. If set to 1, the ruler is displayed in the Report Designer. If 0, no ruler is displayed.
RULERLINES	N(1,0)	Only used in the report definition record. If set to 1, ruler lines are displayed in the

		Report Designer. If 0, no ruler lines are displayed.
GRID	Logical	If set to .T., objects snap to grid in the Report Designer. Only used in the report definition record.
GRIDV	N(2,0)	Vertical spacing, in pixels, for grid lines in the Report Designer. Only used in the report definition record.
GRIDH	N(2,0)	Horizontal spacing, in pixels, for grid lines in the Report Designer. Only used in the report definition record.
FLOAT	Logical	Used for report objects. If set to .T., object floats relative to the size of nearby fields.
STRETCH	Logical	Applies to shapes and fields. If set to .T. for fields, object stretches vertically to show all data. For shapes, if set to .T., the stretch is relative to the tallest object in the group.
STRETCHTOP	Logical	Applies to shapes only. If set to .T., the shape stretches relative to the height of the band.
TOP	Logical	In the report definition record, normally TOP and BOTTOM are set to .F. If this is a multiple column report and print order is left to right (see Page Setup), then BOTTOM is set to .T. For report objects, this determines whether anobject is fixed relative to the top of the band.
BOTTOM	Logical	In the report definition record, this is set to .T. if the print order is set for left to right. With report objects, this is set to .T. if an object is fixed relative to the bottom of a band. Note that with report objects, either TOP or BOTTOM is .T., never both.
SUPTYPE	N(1,0)	Not used.
SUPREST	N(1,0)	Not used.
NOREPEAT	Logical	Set for report bands and report objects. Seems to always be set to .F. for bands, except for the summary band, where it's set to .T. For report objects, corresponds with the setting of "Print repeated values" in the Print When dialog box for report fields.
RESETRPT	N(2,0)	Not used.

PAGEBREAK	Logical	Used in report bands only. Corresponds to the "Start new group on new page" setting. Again, the summary band setting defaults to .T.
COLBREAK	Logical	Used in report bands only and only respected in multi-column reports. Corresponds to "Start new group in new column" setting. The summary band record has a value of .T.
RESETPAGE	Logical	Used in report bands only. If set to .T., the page number is reset to 1 on new pages.
GENERAL	N(2,0)	Only used for bitmaps (OBJTYPE=17). If this value is 1, the object comes from a General field. If the value is 2, the object comes from a file, such as a BMP file.
SPACING	N(3,0)	Used with labels and report fields. Indicates the spacing between lines.
DOUBLE	Logical	Only used for bitmaps. If this is set to .T., the picture is centered in the field. Defaults to .F.
SWAPHEADER	Logical	Not used.
SWAPFOOTER	Logical	Not used.
EJECTBEFOR	Logical	Not used.
EJECTAFTER	Logical	Not used.
PLAIN	Logical	Apparently not used, but summary band records always have this value set to .T.
SUMMARY	Logical	Not used.
ADDALIAS	Logical	Appears to always be set to .T. in the report definition record, but the Microsoft reports say it's not used.
OFFSET	N(3,0)	Used with report fields, shapes, and bitmaps. I'm not sure what function it performs with bitmaps. For fields, it appears that 0 = Align left, 1 = Align right, and 2 = Align center. For shapes, it appears to be a rounding value.
TOPMARGIN	N(3,0)	Not used.
BOTMARGIN	N(3,0)	Not used.
TOTALTYPE	N(3,0)	Applies to report fields and report variables. This is the type of calculation done. The values are: 0 = Nothing

		1 = Count 2 = Sum 3 = Average 4 = Lowest 5 = Highest 6 = Standard deviation 7 = Standard variance
RESETTOTAL	N(2,0)	Applies to report fields and report variables. The default value is 1, at the end of the report, for fields. The full range of fixed values is: 1 = At end of report 2 = At end of page 3 = At end of column Groups are indicated as 5 + group number. So a 6 would indicate that the values are reset after the first group, a 7 the second group, and so on.
RESOID	N(3,0)	Not used. Completely undocumented in 60FRX1.
CURPOS	Logical	Not documented. Seems to always be set to .F. in the report definition record.
SUPALWAYS	Logical	Used with report objects. If set to .F., values repeated from the previous record are suppressed.
SUPOVFLOW	Logical	Used with report objects. If set to .T., the item will be printed when detail overflows to a new page or column. Maps to the check box of the same caption in the Print When dialog box in the Report Designer.
SUPRCOL	N(1,0)	Used with report objects. Defaults to a value of 3, indicating that this object should be printed in the first whole band of a new page or column. Maps to a dialog box in the Print When window in the Report Designer.
SUPGROUP	N(2,0)	Used with report objects. Indicates when an object should be printed when the group changes. Uses the same values as RESETTOTAL, and 0 indicates that the object is always printed.

SUPVALCHNG	Logical	Used with report objects. The Microsoft documentation says that this flag is used to indicate whether the object shouldn't be printed on every band. I've been unable to confirm this.
SUPEXPR	Memo	Used with report objects. The "Print When" expression.
USER	Memo	Can be used for user comments. Untouched by VFP itself.

Acknowledgements

I'd like to express appreciation to Markus Egger for his work in exploring this topic in conjunction with EPS Software's GenRepoX freeware utility. Markus' research into the FRX is available at www.eps-software.com, and I doubt that I'd have gotten very far without the "jumpstart" his research provided.

Conclusion

Understanding the basics of what's stored in the Visual FoxPro report tables is only the beginning. Several people and firms market and distribute utilities that create report files from scratch or modify existing ones, so I know I'm not the first person to dig this deep into these files.

The trick, and a lovely one it would be, would be for someone to completely reverse-engineer the way that Visual FoxPro interprets the report files at runtime. I've made some headway in that direction; the results can be found as FRX2Word, a shareware classlib available at the Universal Thread, VFUG, and CompuServe.

A complete reverse-engineering would enable developers to export VFP reports to different formats, and add tricks and useful stuff to existing reports, and would make a lot of people's lives a lot easier. It's unlikely that Microsoft will be adding export capabilities to the VFP report engine anytime soon, so I guess it's up to us!